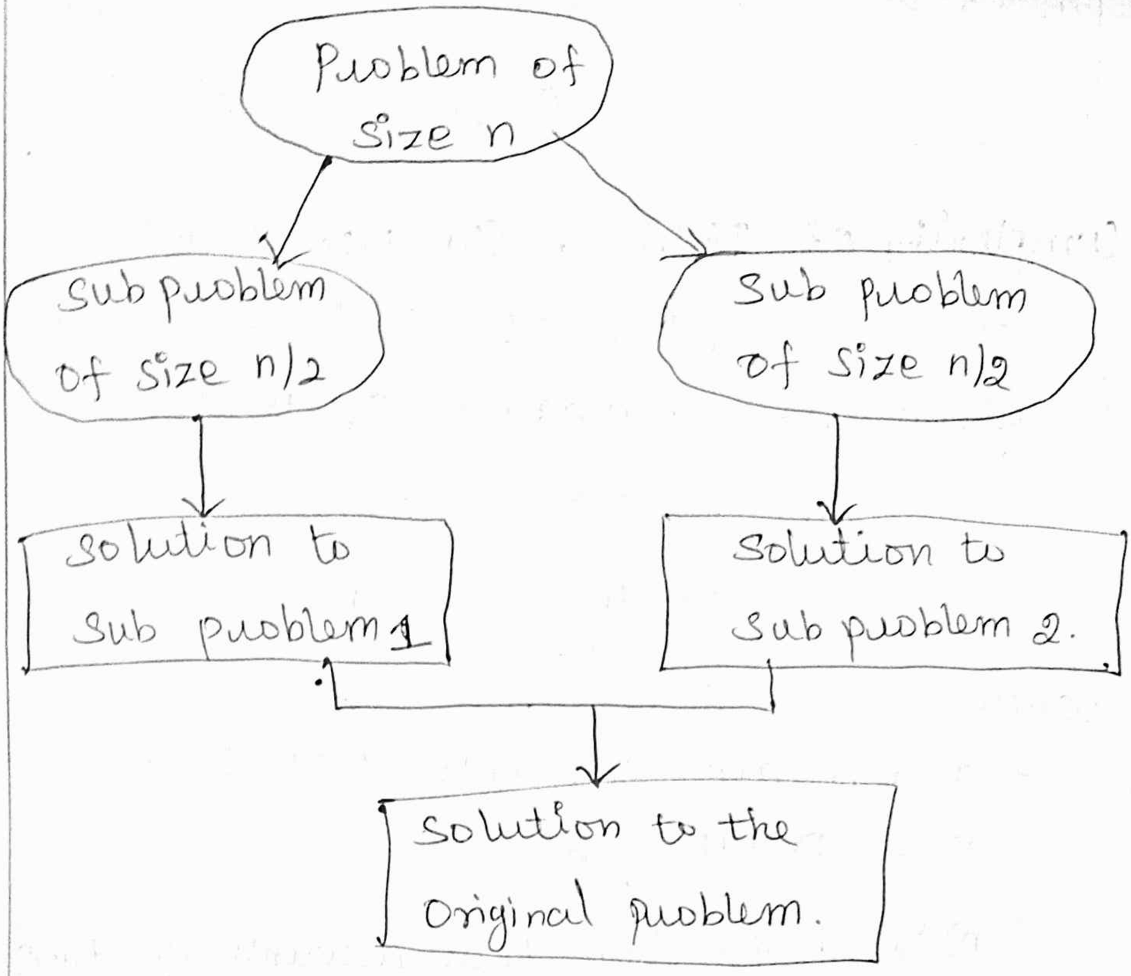
PART-II

DIVIDE AND CONQUER METHODOLOGY:

Definition:

It split the I/P into 'K' subsets $1 < K \leq n$ yielding 'K' subproblems. These subproblems must be found to combine subsolutions into the solution of the whole.



General method:

→ Divide the problem 'P' into $P_1, P_2 \ldots P_k$ sub problems until P is smaller one.

→ Solve the problem using Recursive Alg.

→ The solution to the problem using Recursive algorithm.

Ex: Computing sum of 'n' numbers:-

→ If $n>1$, we can divide the problem into 2 instances.

→ Compute the sum of first $(n/2)$ numbers.

→ Compute the sum of remaining $(n/2)$ "

→ Once the instance are computed, add their values to get the sum of original problems

$$\boxed{a_0 + a_1 + \dots a^{n-1} = (a_0 + a_1 \dots a[n/2 - 1] + 1) \quad (a[n/2) + \dots a^{n-1}))}$$

Complexity of Divide & Conquer Technique:

The complexity of divide & conquer is given by recurrence of the form.

$$T(n) = \begin{cases} T(1) & n = 1 \\ aT\left(\frac{n}{b}\right) + f(n) & n > 1 \end{cases}$$

where,

- → a & b are constants $a \geq 1, b \geq 1$
- → n is power of b.

→ f(n) is a function that accounts for time spent on dividing the problem into smaller ones & on combining their Solutions.

→ T(n) depends on the values of the constant a & b and order of growth of the the function f(n).

Ex:

$a = 2, b = 2.$

Let $T(1) = 2$ & $f(n) = n$

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

$$= 2\left(2T\left(\frac{n}{4}\right) + \left(\frac{n}{2}\right)\right) + n$$

$$= 4T\left(\frac{n}{4}\right) + 2n$$

$$= 4\left(2T\left(\frac{n}{8}\right) + \left(\frac{n}{4}\right)\right) + 2n$$

$$= 8T\left(\frac{n}{8}\right) + 3n$$

$\vdots$

$$T(n) = 2^i T\left(\frac{2}{2^i}\right) + in$$

$$\boxed{\therefore T(n) = n \log_2^n + 2n.}$$

Ex: of Divide & Conquer Technique:-

1. Merge Sort 2. Quick Sort
3. Binary search 4. Binary Tree Traversal.
5. Multiplication of large integers.
6. Strassen's matrix multiplication.